

Contenido

Introducción	1
Maryline CHETTO y Audrey QUEUDET	
Capítulo 1. Informática en tiempo real	3
Audrey QUEUDET	
1.1. La terminología “tiempo real”	4
1.1.1. Características y propiedades de los sistemas en tiempo real	4
1.2. Clasificación de los sistemas en tiempo real	6
1.2.1. Duro, suave o firme: ¿cuáles son las restricciones de tiempo?	6
1.2.2. <i>Event-Triggered</i> versus <i>Time-Triggered</i> : ¿qué programación?	7
1.3. Ejemplos típicos de sistemas en tiempo real	8
1.3.1. Sistemas de aeronáutica	8
1.3.2. Sistemas multimedia	11
1.3.3. Sistemas médicos	13
1.4. Sistemas operativos en tiempo real: ¿cuáles son sus características específicas?	14
1.4.1. Programación de tareas	14
1.4.2. Prioridad básica	15
1.4.3. Asignación dinámica de memoria	15
1.4.4. Gestión de las interrupciones	16
1.4.5. Configuraciones de hardware	17
1.5. Ejemplos de sistemas operativos en tiempo real para sistemas integrados	17
1.5.1. FreeRTOS	17
1.5.2. Zephyr	19
1.5.3. μ C/OS-III	20

1.5.4. Keil RTX	22
1.5.5. Nucleus	23
1.5.6. ThreadX-lite	24
1.5.7. Contiki	25
1.6. Conclusión	26
Capítulo 2. Fundamentos de la programación en tiempo real	29
Audrey QUEUDET	
2.1. Caracterización y modelos de tareas en tiempo real	30
2.1.1. Definiciones de los términos	30
2.1.2. Modelos de tareas	32
2.2. Análisis de programación	34
2.2.1. Definiciones básicas relacionadas con el análisis de programación	35
2.2.2. Enfoques para el análisis de programación	37
2.3. Problema de programación de un solo procesador	38
2.3.1. Tipología de los algoritmos de programación	38
2.3.2. Propiedades de los algoritmos de programación	40
2.3.3. Complejidad de los algoritmos de programación	41
2.4. Programación de tareas periódicas	41
2.4.1. Programación con prioridades fijas	42
2.4.2. Programación con prioridades dinámicas	46
2.5. Servidores de tareas aperiódicas	53
2.5.1. Método de ejecución en procesamiento de fondo	54
2.5.2. Enfoques basados en servidores con prioridades fijas	54
2.5.3. Enfoques basados en servidores con prioridades dinámicas	55
2.6. Conclusión	56
Capítulo 3. La energía ambiental al servicio de los sistemas embebidos	57
Maryline CHETTO	
3.1. ¿Por qué recuperar energía del medio ambiente?	58
3.1.1. Una tecnología en constante evolución	58
3.1.2. Terminología	58
3.1.3. La huella ecológica	60
3.1.4. Miniaturización	60
3.1.5. La duración de la autonomía	61
3.1.6. Ahorro de energía	62
3.1.7. En resumen	62

3.2. Amplia gama de aplicaciones	63
3.2.1. Las infraestructuras	64
3.2.2. La salud.	64
3.2.3. El deporte.	65
3.2.4. Actividades de ocio.	65
3.2.5. Bienestar	65
3.2.6. Domótica y seguridad	66
3.2.7. Vehículos de transporte	66
3.3. Fuentes de energía explotables	66
3.3.1. Diversidad e infinito	66
3.3.2. Fuentes mecánicas	67
3.3.3. El calor	69
3.3.4. La luz	69
3.3.5. El cuerpo humano.	71
3.3.6. Radiación electromagnética.	72
3.4. Almacenamiento de energía	73
3.4.1. Introducción	73
3.4.2. Baterías químicas recargables.	74
3.4.3. Supercapacitores	75
3.4.4. Baterías recargables en estado sólido	77
3.5. Implementación de un sistema autónomo	78
3.5.1. Componentes de un sistema autónomo	78
3.5.2. Evaluación de las necesidades energéticas	79
3.6. Principios de funcionamiento actuales	79
3.6.1. Sistemas de almacenamiento y uso.	80
3.6.2. Sistemas de pulso de energía	80
3.7. Conclusión	81

Capítulo 4. Autonomía energética y programación en tiempo real.

Maryline CHETTO

4.1. Tiempo y energía: una doble limitación	84
4.2. Descripción de un sistema embebido autónomo	85
4.2.1. Arquitectura de hardware	85
4.2.2. Arquitectura de software.	86
4.2.3. El modelo RTEH	87
4.2.4. Terminología.	89
4.3. Principales resultados teóricos.	90
4.3.1. Debilidades de EDF	90
4.3.2. Ejemplo ilustrativo	91

4.3.3. Los puntos fuertes de EDF	92
4.3.4. Necesidad de previsión.	93
4.4. Conceptos.	94
4.4.1. Conceptos relacionados con el tiempo.	94
4.4.2. Conceptos relacionados con la energía	95
4.5. Programador ED-H	97
4.5.1. Principios.	97
4.5.2. Descripción algorítmica	98
4.5.3. Desempeño.	99
4.5.4. Clarividencia.	100
4.5.5. Condición de programación.	100
4.5.6. Ejemplo ilustrativo	101
4.6. Otra solución de programación: LSA	102
4.6.1. Suposiciones	103
4.6.2. Principios.	103
4.6.3. Condición de programación.	105
4.6.4. Ejemplo ilustrativo	105
4.7. Cerraduras tecnológicas	106
4.7.1. Estimación de la energía recuperada.	106
4.7.2. Evaluación de la energía almacenada	107
4.7.3. Prueba de programación	107
4.8. Conclusión	108
Conclusión	111
Maryline CHETTO y Audrey QUEUDET	
Bibliografía	113
Índice alfabético	121